WE CLAIM:

1. A method for responding to a spurious timeout, comprising:

adjusting congestion state values;

maintaining a data flow on a network in accordance with the adjusted congestion state values; and

re-transmitting previously transmitted data when the previously transmitted data has been deemed to be lost on the network.

2. A method according to Claim 1, where the adjusting the congestion state values includes:

restoring a slow-start threshold value;

setting a pipe value; and

re-setting an initial value of a congestion window.

3. A method according to Claim 2, wherein the slow-start threshold is a value of usable bandwidth detected prior to the timeout.

4. A method according to Claim 2, wherein setting the pipe value includes adding a maximum segment size capable of being sent by a sending host to the difference between the maximum sequence number sent so far and the lowest sequence number that is not yet acknowledged.

5. A method according to Claim 2, wherein the re-setting the initial

value of the congestion window includes setting the congestion window to be double that of a maximum data segment size.

6. A method according to Claim 2, wherein maintaining a data flow in accordance with the adjusted congestion state values includes:

transmitting a data packet;

receiving an acknowledgement; and

re-setting the congestion window by adding the maximum data segment size capable of being sent by the sending host.

7. A method according to Claim 2, wherein re-transmitting previously transmitted data when the previously transmitted data has been deemed to be lost on the network includes re-transmitting previously transmitted data when three duplicate acknowledgements are received by a sending host.

8. A method according to Claim 7, further comprising implementing a slow-start recovery process.

9. A method according to Claim 7, further comprising implementing a slow-start recovery process, which includes:

readjusting the pipe value; and

re-setting the size of the congestion window in accordance with a pattern of received acknowledgements.

10. A method for responding to a spurious timeout on a network, comprising:

restoring congestion state values, including setting a limit of data that a sending host can send over the network before receiving an acknowledgement;

maintaining a data flow from the sending host; and

resetting, upon receiving an acknowledgement, the limit of data that the sending host can send over the network before receiving an acknowledgement.

11. A method according to Claim 10, wherein restoring congestion state values includes:

restoring a threshold value of available bandwidth prior to the spurious timeout; and

adjusting an estimate of data outstanding on the network prior to the spurious timeout.

12. A method according to Claim 10, wherein the limit of data that the sending host can send over the network in before receiving an acknowledgement is set to twice the maximum data segment size that the sending host can send.

13. A method according to Claim 10, wherein the limit of data that the sending host can send over the network before receiving an acknowledgement is re-set, upon receiving an acknowledgement, by adding the maximum data segment size that the sending host can send.

14. A method according to Claim 10, further comprising re-transmitting data when data previously transmitted over the network is confirmed to be lost on the network.

15. A method according to Claim 14, wherein data previously transmitted over the network is confirmed to be lost on the network upon receiving three duplicate acknowledgements.

16. A method according to Claim 9, further comprising maintaining a data flow according to a slow-start recovery process.

17. A computer-readable medium having at least one instruction that, upon detecting a timeout on a network, causes at least one processor to:

adjust congestion state values;

maintain a data flow on the network; and

re-transmit previously transmitted data when the previously transmitted data is determined to be lost on the network.

18. A computer-readable medium according to Claim 17, wherein the at least one instruction to adjust congestion state values includes at least one instruction that causes at least one processor to:

adjust an estimate of an amount of data outstanding in the network to one maximum segment size capable of being sent by the sending host plus the difference between the maximum sequence number sent so far and the lowest sequence number that is not yet acknowledged.

19. A computer-readable medium according to Claim 17, wherein the at least one instruction to adjust congestion state values includes at least one instruction to:

limit an amount of data that a sending host can send before receiving an acknowledgement to be twice a maximum data segment size capable of being sent by the sending host.

20. A computer-readable medium according to Claim 19, wherein the at least one instruction to maintain a data flow on the network includes at least one instruction to:

increase the amount of data that the sending host can send before receiving an acknowledgement by the maximum data segment size capable of being sent by the sending host.

21. A computer-readable medium according to Claim 17, wherein the at least one instruction to re-transmit previously transmitted data when the previously transmitted data is determined to be lost on the network includes at least one instruction that causes at least one processor to:

initiate slow-start processing.

22. An apparatus for spurious timeout recovery, comprising:

a transmitter to transmit data packets;

a transmission timer to detect a spurious timeout; and a response processor to maintain a data flow until data is confirmed to be lost on a network.

23. An apparatus according to Claim 22, wherein the response processor is to:

adjust congestion state values;

maintain a data flow on a network in accordance with the adjusted congestion state values; and

re-transmit previously transmitted data when the previously transmitted data has been deemed to be lost on the network.

24. An apparatus according to Claim 23, wherein to adjust congestion state values is to:

set a limit to the amount of data that a sending host can send before receiving an acknowledgement to be twice the size of a data segment that the sending host can send.

25. An apparatus according to Claim 23, wherein to maintain a data flow on the network in accordance with the adjusted congestion state values is to:

reset, upon receiving an acknowledgement, a limit to the amount of data that a sending host can send before receiving an acknowledgement by adding the size of a data segment that the sending host can send; and

transmitting data on the network.

26. An apparatus according to Claim 23, wherein to re-transmit previously transmitted data when the previously transmitted data has been deemed to be lost on the network is to re-transmit the previously transmitted data upon receiving three duplicate acknowledgements.

27. An apparatus according to Claim 27, wherein the apparatus is to further process a slow-start recovery.

28. A processor, comprising:

means for adjusting congestion state values;

means for maintaining a data flow on a network in accordance with the adjusted congestion state values; and

means for re-transmitting previously transmitted data when the previously transmitted data has been deemed to be lost on the network.

29. A processor according to Claim 28, wherein the means for maintaining the data flow on the network in accordance with the adjusted congestion state values re-sets, upon receiving an acknowledgement, a limit to the amount of data that a sending host can send before receiving an acknowledgement by adding the size of a data segment that the sending host can send and continues to transmit data on the network.

30. A processor according to Claim 28, wherein the means for re-transmitting previously transmitted data re-transmits the previously transmitted data upon

receiving three duplicate acknowledgements.